GRASPING THE FUTURE: JOINT 3D HAND–OBJECT FORECASTING FROM INTERNET VIDEOS

Anonymous authors
Paper under double-blind review

ABSTRACT

To interact with an object, we need to anticipate both how it will move and how our hands should move with it. Human videos capture these relationships across everyday activities, offering a natural source for learning to predict future interactions. Learning from them requires recovering motion from completed interactions and using it to anticipate what could happen next. We present *Grasping-the-Future*, a unified generative model of joint 3D hand-object motion, including wrist motion, finger articulation, and object position and orientation. Through masked spatiotemporal completion and flow matching, we train a single model to forecast future interactions, predict hand motion given object motion or vice versa, generate both trajectories toward a desired final object pose, and complete partially observed interactions. We explicitly model hand-object relations and incorporate contact constraints to help preserve grasp relationships during motion. To obtain 3D supervision from internet videos, we develop a scalable pipeline that combines pretrained perception models with joint geometric optimization to reconstruct completed interactions. We combine these recovered trajectories with annotated interaction datasets to learn future motion from partial observations. We demonstrate strong performance across all tasks, outperforming recent state-of-the-art models. Moreover, we show geometric and physical plausibility of the predicted trajectories by retargeting them to a robotic embodiment. Videos are available at https://gtficlr2027.github.io.

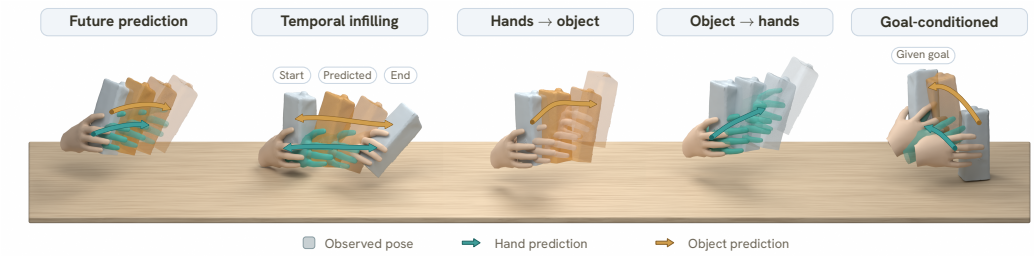


Figure 1: **Unified model for different types of hand-object prediction.** *Grasping-the-Future* forecasts future motion, fills temporal gaps, predicts either entity from the other's motion, and generates motion toward a given goal. Neutral poses show observations; teal and amber arrows indicate hand and object prediction, respectively. Fading poses show successive times. For visualization, forecasting snapshots are spaced apart and cross-entity predictions are contact-refined; arrows are schematic.

1 INTRODUCTION

"Life can only be understood backwards; but it must be lived forwards."
— Søren Kierkegaard

To interact with an object, we need to anticipate how our actions will affect it. Placing a mug on a shelf is a simple everyday action, but predicting how it will unfold requires understanding how the hand and mug move together. The fingers must hold the mug as it is lifted and release it only once it rests on the shelf. The way the mug is grasped affects how it can be carried, and the position of the shelf affects where the hand needs to move. From the eyes of the human manipulating, parts

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

of the hand or the object may be occluded, and several plausible future motions may be possible. Anticipating the interaction therefore requires predicting the motion of both the hand and the object, while preserving firm hand-object contact as the object moves through the 3D world.

Human videos are a natural source for learning these predictions, showing how people handle objects across many everyday situations. Learning from them involves two steps: first, extracting hand and object motion from completed interactions, and then using these examples to learn how an interaction might continue from a partial observation. These steps echo Kierkegaard's distinction: a recorded video allows us to look back and understand what happened, while acting requires anticipating what comes next. Image-based predictions, such as generated videos, can leave relative depth and contact ambiguous. Predicting in 3D makes the spatial relationships between the hands, objects, and scene explicit: where the fingers touch the mug, how it is oriented as it is carried, and where it comes to rest on the shelf. This geometric description also provides an unambiguous reference for later re-targeting the predicted motion to a different embodiment like a robot.

Our key insight is that a *single* model can learn these different forms of prediction: predicting an object's trajectory given the hand's motion, predicting the hand's trajectory given the object's motion, generating both trajectories given a desired final object pose, and forecasting or completing an interaction from partial 3D observations. People reason this way intuitively: watching a hand move suggests where an object will go, while imagining where an object should end up suggests how the hand needs to move. We present ***Grasping-the-Future***, a unified generative model of joint 3D hand-object motion. The same model forecasts an interaction from a short observed context, predicts hand trajectories given object motion or object trajectories given hand motion, generates motion toward a desired object pose, and fills gaps between observed poses (Figure 1). Across these tasks, we must handle observations of different entities and times. We bring these tasks together through masked spatiotemporal completion: we reveal different parts of a hand-object trajectory to the prediction model, and a flow-matching objective learns to generate the missing pieces of the trajectory. The generated trajectories include wrist motion, finger articulation, and object position and orientation. Since plausible hand and object trajectories can still drift apart, we use explicit hand-object relations and contact constraints to keep them moving together during a grasp.

Learning this predictive model requires 3D hand-object supervision that large-scale internet videos do not directly provide. We therefore develop a scalable pipeline that combines pretrained perception models with joint geometric optimization to extract hand-object trajectories from complete monocular videos. The pipeline separates camera motion from object motion, refines the alignment between the hands and object using contact and temporal constraints, and filters unreliable reconstructions. We extract over $62{,}000$ trajectories from internet videos and combine them with existing annotated hand-object datasets in a common 3D representation, accounting for differences in annotation quality during training. We evaluate our reconstructions against ground-truth annotations on EPIC-Contact and recover hand trajectories that closely follow the reference motion. We use the reconstructed trajectories as pseudo-ground truth to learn future motion from short observed portions of each interaction.

We evaluate *Grasping-the-Future* across forecasting and conditional generation tasks on datasets with 3D annotations, including OakInk2 (Zhan et al., 2024) and HOT3D (Banerjee et al., 2025), and on held-out internet videos using our extracted trajectories as pseudo-ground truth. Our best-of-20 forecasts outperform the evaluated state-of-the-art methods in both wrist and object ADE and FDE, reducing object FDE from 127.6 to 102.8 mm (Table 2). We also demonstrate that these predictions can guide robot manipulation: by re-targeting the predicted human motion to simulated robot hands with SPIDER (Pan et al., 2026), we enable the hands to manipulate objects along the forecast trajectories for different types of manipulation tasks.

In summary, our contributions are threefold. We develop a unified generative model of 3D hand-object motion that supports forecasting, cross-entity prediction, goal-conditioned generation, and temporal completion through a common masking formulation. We propose a scalable pipeline for reconstructing temporally aligned hand-object trajectories from internet videos and using them as supervision for learning future interactions. We incorporate explicit hand-object relations and geometric contact constraints into trajectory generation, producing structured 3D predictions that we evaluate through re-targeting and object manipulation in simulation.

2 Related Work

Reconstructing hand-object interactions from video. Recent methods recover 3D hand-object geometry and motion from monocular videos by combining visual observations with geometric and learned priors (Hasson et al., 2021). Joint priors over hand configurations and object shape support reconstruction and grasp synthesis (Ye et al., 2024), while temporal attention helps recover coherent hand trajectories (Ye et al., 2026a). Other methods reconstruct bimanual interactions using Gaussian splatting (On et al., 2025) or use novel-view synthesis priors to recover object geometry that is not visible in short clips (Wang et al., 2025b). Hand priors also support joint hand-object pose estimation from a single egocentric image (Bansal et al., 2026). Recent work recovers world-space hand and object motion from egocentric videos with known object templates and supports hand-guided interaction synthesis (Ye et al., 2026b). We build on these advances to extract supervision from completed human interactions. Our focus is on using the recovered trajectories to learn how an interaction might continue from a partial observation.

Interaction forecasting and robot manipulation. Interaction forecasting asks how an observed interaction is likely to continue. Human videos show how people reach for, grasp, and move objects. Prior work predicts hand trajectories and interaction hotspots from egocentric observations (Liu et al., 2022), including uncertainty-aware 3D hand forecasts (Bao et al., 2023). Recent methods incorporate language queries into hand trajectory prediction (Bao et al., 2025) and connect interaction reasoning with 3D hand motion prediction (Chen et al., 2026b). Complementary approaches predict future object poses (Soraki et al., 2026) or 3D object point trajectories (Bharadhwaj & Jangir, 2026; Zhang et al., 2026). Object motion can also be forecast by conditioning on hand motion (Thakkar et al., 2025), while full-body human-object forecasting combines diffusion with physics-informed correction (Xu et al., 2023). Dense 2D point trajectories also support motion forecasting in unconstrained videos (Thakkar et al., 2026). These representations provide useful cues for robot manipulation. Prior work translates predicted human interaction plans into robot actions (Bharadhwaj et al., 2024a), combines predicted point tracks with geometric transfer and a residual policy (Bharadhwaj et al., 2024b), or trains policies conditioned on generated human videos (Bharadhwaj et al., 2025). We jointly model articulated hand motion and object pose, capturing how the fingers, wrist, and object move together. We evaluate their use for manipulation by forecasting human interactions in a robot's scene and retargeting the predicted motion for open-loop execution.

Controllable motion synthesis. A complementary line of work, often motivated by graphics and animation, generates plausible motion from text, object geometry, reference grasps, or goal poses. These methods emphasize satisfying the supplied conditions, often without observations of the surrounding scene. Prior work synthesizes dynamic grasps from a grasp reference and a desired object pose using physics-based control (Christen et al., 2022). Bimanual manipulation datasets also provide models for completing task primitives (Zhan et al., 2024). Text-conditioned methods generate hand-object motion through contact-guided diffusion (Cha et al., 2024) or masked latent flow matching (Chen et al., 2026c), while other work separates global motion from local hand articulation using latent diffusion and a grasp VAE (Li et al., 2025a). Some methods ground generated motion through initial human-object poses and sparse object waypoints (Li et al., 2024). Masks over frames and joints also support sparse or partial motion constraints (Cohan et al., 2024). We ground our predictions in partial observations of a hand-object interaction and the surrounding scene geometry. In *Grasping-the-Future*, one model forecasts its continuation, predicts either entity from the other's motion, generates both trajectories toward a desired object pose, and completes missing portions of an interaction while modeling hand-object contact.

3 Method

We develop *Grasping-the-Future* to predict how hands and objects interact with each other in 3D. We represent an interaction as a trajectory of hand and object poses. We observe a part of this trajectory, and our goal is to generate the rest. By changing which parts we observe, we use the same model to forecast the future, to generate the hand trajectory from a given object trajectory and vice versa, to generate the motion that reaches a given final pose, and to fill a gap in the middle of a trajectory. We first describe this representation and the prediction model and then introduce the losses that help maintain hand-object contact. Finally, we describe how we obtain 3D hand-object reconstructions from internet videos to create a dataset for training the model.

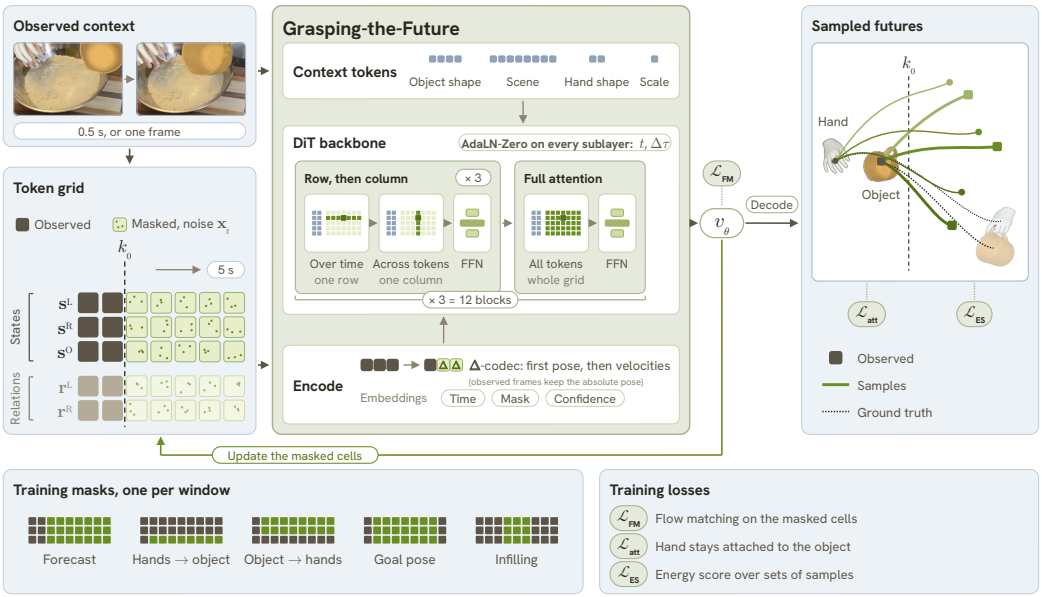


Figure 2: **Model overview.** The top row shows inference. An interaction is a grid of tokens with one row per state $\mathbf{s}$ and relation $\mathbf{r}$ and one column per frame. Observed cells are given and the rest start as noise. A transformer that alternates attention along rows, down columns and over the whole grid predicts the flow velocity v_θ for the noisy cells, with the context tokens (blue) as extra keys and values. Ten Euler steps turn the noise into a sample, which is decoded into hand and object trajectories. Several samples for the same input are shown against the ground truth (dotted). The bottom row shows training. One observation mask is drawn per window, so every task trains the same model with different cells observed. Flow matching supervises the velocity and auxiliary losses act on the decoded trajectories (Section 3.3).

3.1 Problem Formulation

Representation. We segment each recorded interaction into windows of T consecutive frames. A window contains up to two hands and one rigid object, the three entities we model. At frame k, each hand $e \in \{L, R\}$ has a state $\mathbf{s}_k^e \in \mathbb{R}^{99}$ made of the wrist position and rotation and the 15 joint rotations of the MANO hand model (Romero et al., 2017). The object state, $\mathbf{s}_k^O \in \mathbb{R}^9$, is made of its position and rotation. Every rotation is in the continuous 6-D form (Zhou et al., 2019). All poses are expressed in the camera coordinates of the last observed frame, which removes camera motion from the window. We also subtract from every position the 3D point halfway between the observed wrists and the object at that frame, so that each window is centered on the interaction. The states of a window form $\mathbf{X}$. We also give the model a static context $\mathbf{c}$ with the object mesh and its scale, the MANO shape of each hand, the gravity direction when available, and a point cloud of the scene.

Input and output. We provide the model with a mask $\mathbf{m} \in \{0, 1\}^{3\times T}$, where each row represents an entity and each column corresponds to a frame, indicating which states within a window we supply to the model and which it should generate. A cell is considered 1 when we provide the model with the entity's state at that frame, while $\bar{\mathbf{m}}$ indicates the cells that need to be generated. We also provide the model with which entities are present at each frame. The model learns a distribution over the generated states, $\mathbf{X}_{\bar{\mathbf{m}}} \sim p_\theta(\cdot \mid \mathbf{X}_{\mathbf{m}}, \mathbf{c})$, and draws samples from it. The tasks differ only in the mask:

- **Forecasting**, the main task. The first frames of the window are observed and everything after them is generated.
- **Hands → object and object → hands.** One entity is observed throughout the window and the other only at the first frame.
- **Goal-conditioned generation.** Only the first frame is observed, together with the final state of the object, the hands, or both, and the model generates the motion toward that final state.

- **Temporal infilling.** A sequence of frames at the beginning and at the end of the window is observed for every entity, while the missing segment in the middle is generated.

3.2 Forecasting as Masked Spatiotemporal Completion

We generate the masked cells with a flow-matching transformer over a grid of tokens (Figure 2).

Relation tokens. To make the model generate the hand-object relation directly, rather than leave it to fall out of two separately generated poses, we add a *relation token* for each hand that describes the hand relative to the object. Each frame then has five tokens, the three states and the two relation tokens. The relation token holds the wrist pose expressed in the object's coordinates and the distance from each of the five fingertips to the object surface. Let $(R_k^O, \mathbf{p}_k^O)$ denote the object's rotation and position at frame k, $(R_k^e, \mathbf{p}_k^e)$ the wrist's rotation and position, $(\mathbf{u}_{f,k}^e)$ the position of fingertip f, and $(\mathcal{S}_k)$ the object surface. We define the token as

$$\mathbf{r}_k^e = \Big[\; R_k^{O\top}\big(\mathbf{p}_k^e - \mathbf{p}_k^O\big),\;\; R_k^{O\top} R_k^e,\;\; \big\{\operatorname{dist}\big(\mathbf{u}_{f,k}^e, \mathcal{S}_k\big)\big\}_{f=1}^{5} \;\Big] \in \mathbb{R}^{14}. \tag{1}$$

If the hand or the object is not observed in a frame, we do not give the model that frame's relation token either, since it would reveal a pose the model has to generate. The model then generates the relation token along with the pose.

Δ-codec and flow matching. We represent motion using a temporal-difference encoding, which we call the Δ-codec. Within each uninterrupted interval in which an entity is present, we store the first state in standardized absolute form and subsequent states as changes per unit time. Writing $\mathbf{s}_k$ for the state at time τ_k, we encode these changes as

$$\mathbf{z}_k = \frac{\mathbf{s}_k - \mathbf{s}_{k-1}}{(\tau_k - \tau_{k-1})\,\boldsymbol{\sigma}_v}. \tag{2}$$

Here, $\boldsymbol{\sigma}_v$ contains per-channel velocity scales estimated from training data. Stationary states have zero difference tokens, and dividing by elapsed time accommodates different frame rates. A difference token is observed only when both adjacent states are observed. We also supply absolute observed states as separate inputs, allowing the model to use observations on either side of a gap.

We recover absolute states by integrating the changes from each interval's initial state. At inference, we correct generated gaps using their observed boundary poses. For gaps bounded on both sides, we interpolate translation corrections linearly in elapsed time and rotation corrections along geodesics on $\mathrm{SO}(3)$. We preserve all observed states exactly (Appendix A).

We train the masked tokens with flow matching (Lipman et al., 2023). With $\mathbf{x}_1$ their clean encoded values, $\boldsymbol{\epsilon}$ Gaussian noise and $\mathbf{x}_t = (1-t)\boldsymbol{\epsilon} + t\mathbf{x}_1$ at a flow time $t \sim \mathcal{U}(0,1)$, the network $v_\theta(\mathbf{x}_t, t; \mathbf{X}_\mathbf{m}, \mathbf{c})$ regresses the velocity $\mathbf{x}_1 - \boldsymbol{\epsilon}$ on the masked tokens, while the observed tokens enter without noise. We call this loss $\mathcal{L}_{\mathrm{FM}}$.

Architecture and sampling. We embed each token with a layer for its type. We then add a sinusoidal encoding of elapsed physical time and learned embeddings for token type, observation status, and confidence. The confidence says how reliable the token's label is. It combines a fixed score for the data source, higher for annotated datasets than for trajectories extracted from video (Section 3.4), with simple checks on the trajectory near that frame, for example how smooth it is. This lets the model rely less on noisy labels. The tokens pass through a transformer built from groups of four blocks. Three attend along time within each row of the grid and then across the tokens of one frame, and the fourth attends over the whole grid. The context $\mathbf{c}$ enters every attention layer as extra key-value tokens, with the object mesh encoded by Uni3D (Zhou et al., 2024) and the scene point cloud, carrying DINOv3 features (Siméoni et al., 2025), by LitePT (Yue et al., 2026). At inference, masked tokens start as Gaussian noise and are denoised in 10 Euler steps of the predicted flow (Appendix A).

3.3 Losses on Decoded Trajectories

Small errors under framewise flow matching can accumulate in decoded trajectories, causing the hand to drift from the object. We therefore supervise hand-object attachment and score complete sampled trajectories.



Figure 3: **Extraction pipeline.** We combine pretrained estimates of hands, objects, depth, and camera motion into a canonical trajectory through four refinement steps (green). We retain only clips that pass all consistency checks. The bar shows how many clips we discard across stages, and the bottom row shows examples rejected by each check.

Attachment loss. We estimate and decode a clean trajectory from a single flow-velocity prediction. For each hand e, we express its wrist position $\mathbf{q}_k^e$, wrist rotation $R_k^{\mathrm{rel},e}$, and fingertip positions $\mathbf{u}_{f,k}^e$ in the object's coordinate frame. We define reference contact frames $\mathcal{C}_e$ as those with at least one fingertip within 5 mm of the object surface. For each hand, we minimize

$$\mathcal{L}_{\mathrm{att}} = \sum_{k \in \mathcal{C}_e} \Big[\|\hat{\mathbf{q}}_k^e - \mathbf{q}_k^e\| + \tfrac{1}{5}\textstyle\sum_f \|\hat{\mathbf{u}}_{f,k}^e - \mathbf{u}_{f,k}^e\| \;+\; \lambda_R\, d_{\mathrm{c}}\big(\hat{R}_k^{\mathrm{rel},e}, R_k^{\mathrm{rel},e}\big) \;+\; \lambda_v\, \|\dot{\hat{\mathbf{q}}}_k^e - \dot{\mathbf{q}}_k^e\| \Big], \quad (3)$$

where hats denote predictions, d_{c} is the chordal rotation distance, and $\dot{\mathbf{q}}_k^e$ is the object-relative wrist velocity. We match the reference relative motion, allowing changes such as regrasping. We also supervise fingertip-to-surface distances and penalize acceleration, jerk, and finger-joint violations beyond their reference values (Appendix A).

Scoring sets of samples. To capture multiple plausible futures, we generate $M = 3$ trajectories X_i from the same observations on a quarter of optimizer steps. We use six Euler steps per sample and backpropagate through all six. We compare these samples with the annotated or reconstructed trajectory Y using the energy score (Gneiting & Raftery, 2007; Pacchiardi et al., 2024; De Bortoli et al., 2025), which balances agreement with the reference against diversity among samples:

$$\mathcal{L}_{\mathrm{ES}} = \mathrm{ES}\big(\{\phi_\ell(X_i)\}_{i=1}^M,\, \phi_\ell(Y)\big) \;+\; \lambda_d\, \mathrm{ES}\big(\{\phi_d(X_i)\}_{i=1}^M,\, \phi_d(Y)\big). \quad (4)$$

The landmark summary ϕ_ℓ records each entity's displacement at one-third, two-thirds, and the end of its generated frames, its final rotation change, and changes in wrist-object offsets at the same three points. The dense summary ϕ_d records positional changes at every generated frame, with $\lambda_d = 0.5$. We score these summaries separately to preserve emphasis on endpoints and rotations. The expected score is minimized when the predicted and reference summary distributions match (Pic et al., 2025).

3.4 Extracting Trajectories from Video

We extract 3D hand and object trajectories from monocular internet videos to expand our training data. Figure 3 summarizes the pipeline; we provide implementation details in Appendix B.

Per-frame estimates. We use Molmo 2 (Clark et al., 2026) to identify manipulation clips from Action100M (Chen et al., 2026a) and locate the manipulated object, then track its mask with SAM 3 (Carion et al., 2025). We estimate MANO hand poses with WiLoR (Potamias et al., 2025), object meshes and poses with SAM 3D (Chen et al., 2025), metric depth with MoGe (Wang et al., 2025a), and camera motion with MegaSaM (Li et al., 2025b).

Trajectory refinement. We compensate for camera motion to express the interaction in a shared world coordinate system. We lift CoTracker3 (Karaev et al., 2025) object tracks into 3D using depth

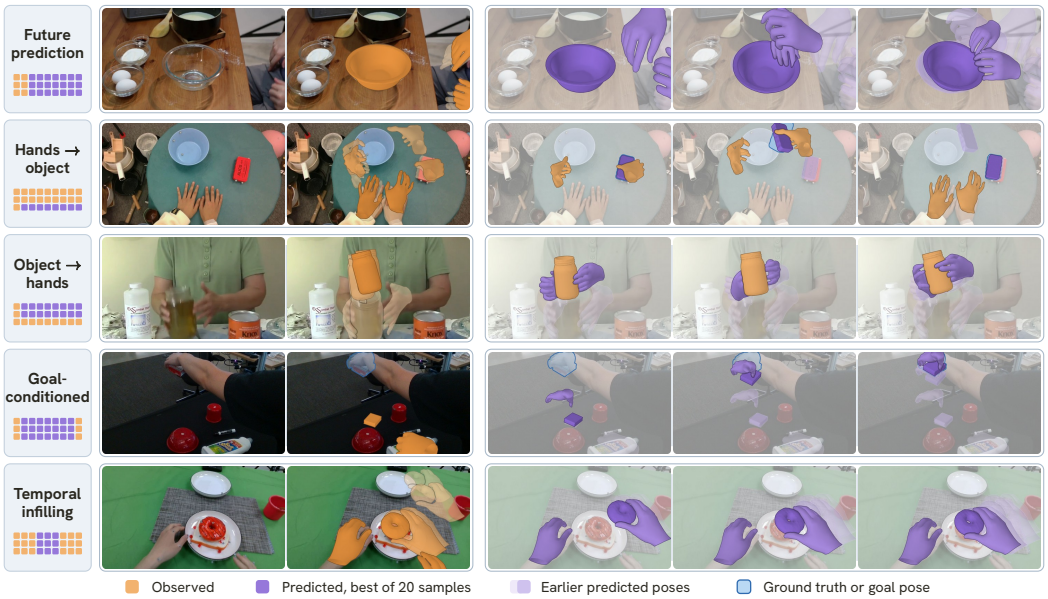


Figure 4: **Qualitative predictions across five tasks.** Each row shows one clip, its observation mask, the input frame with and without the observed motion, and three snapshots of the best of 20 predictions. Mask rows correspond to the two hands and the object. Orange denotes observations and purple predictions; darker orange indicates later observations, while faded purple shows earlier predictions. Blue denotes the reference object pose in the hands-to-object row and the given goal pose. For visualization, we rigidly adjust the forecast hands into contact with the bowl in the first row and one hand out of the box it penetrates in the goal-conditioned row, and we curl the fingers onto the jar in the object-to-hands row and move the reaching hand in its first snapshot onto the jar.

and fit a rigid object trajectory through bundle adjustment with a robust reprojection loss (Huber, 1964). We then jointly refine hand and object poses to match the masks, preserve contact, and penalize interpenetration, abrupt motion, and implausible finger poses, while keeping hand scale fixed. During sustained occlusion with stable contact, we use hand rotation to guide object rotation. We correct depth bias with a transformer trained on corrupted motion-capture trajectories, applying a constant translation along the viewing direction to both the object and grasping hand. Finally, we smooth residual hand jitters (details are in Appendix B.2).

Verification. We check image agreement, contact, interpenetration, temporal consistency, and object size, and use a vision-language model to verify object identity. We retain approximately 12% of candidate clips, assign each an interaction-fidelity score, and estimate gravity when we can identify a supporting surface.

4 Experiments

Our experiments evaluate prediction across seven tasks and its transfer to robotic motion. We compare against non-learned baselines and released forecasting models from prior work, and evaluate the benefit of training on trajectories extracted from internet videos (Section 4.2). We then evaluate how well forecast trajectories transfer to robots (Section 4.3). Additional ablations on the model's losses and inputs are provided in Appendix D.

4.1 Setup

Training. We train on eight annotated hand-object datasets (Chao et al., 2021; Banerjee et al., 2025; Liu et al., 2024; Zhan et al., 2024; Rim et al., 2026; Hampali et al., 2020; Kwon et al., 2021; Wang et al., 2024) and trajectories extracted from internet videos. Together, they provide 74,312 trajectories and 134,661 five-second windows, including 62,562 trajectories and 109,346 windows from internet videos. Each training window receives a task observation pattern sampled uniformly across the five task families, or a random block mask. Training details are provided in Appendix A.

Table 1: **Prediction across seven tasks.** Single-sample and best-of-20 position errors in millimetres, averaged over left and right wrists for hands. "Error at" distinguishes final-frame error from mid-gap error when the final pose is given. "Motion capture only" excludes extracted video data from training. Lower is better; bold and underline indicate the best and second-best results.

	Future prediction		Hands→obj.	Obj.→hands	Temporal infilling		Goal, all given		Goal, hands given		Goal, object given	
Entity	object	hands	object	hands	object	hands	object	hands	object	hands	object	hands
Error at	end	end	end	end	middle	middle	middle	middle	end	middle	middle	end
Stay-put	142.6	<u>145.6</u>	149.0	156.6	51.7	69.3	98.8	127.7	149.0	127.7	98.8	156.6
Constant velocity	627.1	819.3	–	–	–	–	–	–	–	–	–	–
Interpolation	–	–	–	–	38.2	49.0	76.0	92.3	–	92.3	76.0	–
Rigid carry	–	–	108.6	156.4	–	–	–	–	–	–	–	–
Motion capture only, N=1	159.0	185.6	71.6	132.9	39.0	50.0	91.2	101.9	146.1	110.3	96.4	166.9
Motion capture only, N=20	<u>121.8</u>	153.9	<u>50.0</u>	103.7	<u>26.6</u>	<u>35.0</u>	<u>62.6</u>	<u>74.1</u>	<u>107.4</u>	<u>78.2</u>	<u>68.0</u>	137.1
Ours, N=1	145.8	157.4	59.1	<u>98.7</u>	39.5	52.1	89.8	101.6	126.1	110.2	97.4	<u>129.3</u>
Ours, N=20	**102.8**	**113.2**	**36.6**	**75.1**	**23.4**	**32.8**	**56.5**	**70.2**	**91.1**	**73.8**	**61.6**	**91.7**

Benchmark. We evaluate every task on the same 331 five-second windows from diverse scenarios unseen during training: 256 from the annotated datasets and 75 from internet videos, with one window per video. For forecasting, the model observes the first 0.5 seconds and predicts the remaining 4.5 seconds. Additional details are in the Appendix C. We will release the code and data publicly.

Metrics. We measure position error in millimeters at the wrists and object center. Average displacement error (ADE) averages error over generated frames; final displacement error (FDE) measures final-frame error on windows with reference motion. When the final pose is given, we instead report mid-gap error. We report single-sample and best-of-20 trajectory errors, selecting one sample by the lowest ADE across predicted entities. We first average scores within each dataset, then equally across annotated datasets. The combined score weights the annotated-data average and internet-video average equally. Detailed definitions, eligibility criteria, and additional interaction metrics are provided in Appendix C.

Baselines. Stay-put holds the last observed pose, while constant velocity extrapolates observed motion. Interpolation connects observed endpoint poses, and rigid carry maintains a fixed relative pose between the predicted and observed entities. We also compare with the released hand and object forecasters in Table 2. Each released model uses its required inputs, and all predictions are evaluated with our scoring code.

4.2 Prediction Across Tasks

Table 1 compares our model with non-learned baselines across all seven tasks. When predicting one entity from the other's motion, a single sample reduces object FDE from 108.6 mm with rigid carry to 59.1 mm, and hand FDE from 156.4 to 98.7 mm. Best-of-20 achieves lower error than the non-learned baselines in every task column. However, single samples remain less accurate than stay-put for forecasting and interpolation when both endpoint poses are given. Multiple samples therefore improve coverage of the reference motion, while individual predictions can differ from the observed outcome. Figure 4 shows examples across five tasks.

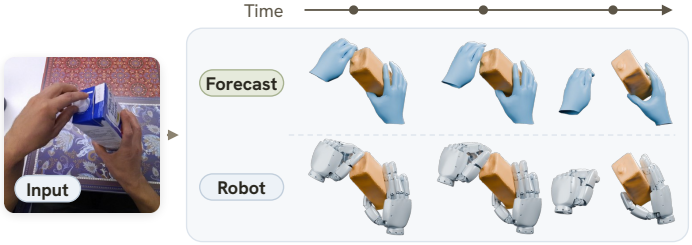
Compared with the model trained on annotated datasets alone, our model has 13 mm lower single-sample forecasting FDE for objects and 28 mm lower FDE for hands (Table 1). Cross-entity prediction improves by 13–34 mm, and prediction of the unconstrained entity under partially specified goals improves by 20–38 mm. Mid-gap errors differ by less than 3 mm when the final pose is observed. The gains are therefore strongest when the endpoint is unknown. We provide reconstruction-quality evaluations in Appendix B.3.

Table 2 compares our model with released forecasting methods. Our model achieves the lowest reported ADE and FDE for both wrists and the object. Wrist FDE is 106–121 mm, compared with 181–193 mm for USST, while object FDE is 102.8 mm, compared with 127.6 mm for ObjectForesight. These results use each method's required inputs and sampling procedure, so the comparison reflects differences across the complete systems.

Comparison with specialist models trained for individual tasks (Table 10) shows that the unified model achieves similar or lower best-of-20 errors on forecasting, cross-entity prediction, and temporal infilling. The goal specialist performs better, particularly for the entity whose final pose is unspecified. These results support using one model across several tasks while identifying partially specified goals as an area where specialization remains beneficial. Detailed specialist results are provided in Appendix E. We also provide ablations of our framework in appendix.

Table 2: **Comparison with different forecasting approaches.** ADE and FDE in millimetres using the listed inputs and sample counts. Deterministic methods use one prediction; stochastic methods report the best of N. Lower is better; bold and underline indicate the best and second-best results.

	Inputs at inference						ADE			FDE		
Method	RGB	Depth	Hands	Object	Text	N	L wrist	R wrist	Object	L wrist	R wrist	Object
USST (Bao et al., 2023)	✓	×	✓	×	×	1	<u>139.5</u>	<u>137.1</u>	–	<u>193.2</u>	<u>181.4</u>	–
ForeHand4D (Prakash et al., 2025)	✓	×	×	×	×	20	253.9	290.0	–	301.2	323.3	–
EgoH4 (Hatano et al., 2025)	✓	×	2D	×	×	1	525.1	568.0	–	569.2	629.7	–
Uni-Hand (Ma et al., 2026)	✓	✓	✓	×	×	20	393.3	373.8	–	412.8	402.2	–
ObjectForesight (Soraki et al., 2026)	×	✓	×	✓	×	20	–	–	<u>72.9</u>	–	–	<u>127.6</u>
MotionForesight (Bharadhwaj & Jangir, 2026)	✓	×	×	mask	×	1	–	–	85.1	–	–	142.9
General Flow (Yuan et al., 2024)	✓	✓	×	pts	✓	20	–	–	153.5	–	–	274.4
MolmoMotion (Zhang et al., 2026)	✓	×	×	pts	×	10	–	–	91.8	–	–	170.0
Grasping-the-Future	×	✓	✓	✓	×	20	**80.4**	**77.1**	**58.9**	**120.6**	**105.9**	**102.8**



Retargeting method	Success rate (%)
Kinematic	14.6
Kinematic + grasp prep	43.7
Dynamic (SPIDER)	**73.8**

Figure 5: **Testing physical plausibility through retargeting.** We test whether simulated robot hands can manipulate the object along the predicted trajectory from *Grasping-the-Future*. The left panel shows the observed frame, our 4.5 s forecast, and simulated replay after SPIDER retargeting (Pan et al., 2026). The table reports success across 103 trajectories in different scenes.

4.3 Testing Physical Plausibility of Forecasting through Retargeting

We assess the physical plausibility of our forecasts by retargeting them to simulated XHands. We test whether the robot hands can manipulate the object and reproduce its predicted motion under the simulator's dynamics. Kinematic retargeting matches the robot's wrist and fingertips to the predicted hand poses. Dynamic retargeting with SPIDER (Pan et al., 2026) further optimizes robot controls in simulation to follow the predicted object motion. We also test kinematic retargeting with SPIDER's grasp preparation but without trajectory optimization. All three methods use the same forecast for each observation. We evaluate 103 examples whose ground-truth hand-object trajectories retarget successfully with SPIDER. A retarget succeeds if the hand manipulates the object with mean position and rotation errors below 100 mm and 0.5 rad relative to the forecast over the prediction interval.

Kinematic re-targeting achieves only 14.6% success, highlighting the difficulty of executing these nontrivial manipulation sequences through wrist and fingertip pose matching alone (Figure 5). Grasp preparation improves success to 43.7%, while dynamic retargeting with SPIDER reaches 73.8%. These results show physical plausibility: after adapting hand motion to the robot embodiment, many forecasted object trajectories can be reproduced through physical interaction in simulation.

5 Conclusion

We presented *Grasping-the-Future*, a unified model for forecasting and completing joint 3D hand-object motion. We combined masked spatiotemporal completion with explicit hand-object relations and contact supervision to predict coordinated motion under different observations and goals. We reconstructed over 62,000 trajectories from internet videos and showed that this supervision improved forecasting beyond training on annotated datasets alone. We also demonstrated that our forecasts could guide object manipulation after re-targeting to simulated robot hands, connecting prediction from human videos to robot execution.

AI USE STATEMENT

We used generative AI tools to assist with writing, polishing, and proofreading the paper, and to generate figures and the video demonstration results. We reviewed all AI-assisted text and rewrote unclear passages. We take full responsibility for the final content of this work, including its text, figures, and claims.

REPRODUCIBILITY STATEMENT

We describe the model architecture, trajectory representation, losses, and training and inference settings in Appendix A. Appendix B details the video extraction pipeline, including pose refinement and quality filtering. Appendix C specifies benchmark selection, evaluation metrics, aggregation, and baseline configurations. We will publicly release our code and data to support reproduction of the results.

REFERENCES

Karim Abou Zeid. JointTransformer: Winning submission to the HANDS'2023 ARCTIC challenge. https://github.com/kabouzeid/JointTransformer, 2023. 24

Prithviraj Banerjee, Sindi Shkodrani, Pierre Moulon, Shreyas Hampali, Shangchen Han, Fan Zhang, Linguang Zhang, Jade Fountain, Edward Miller, Selen Basol, Richard Newcombe, Robert Wang, Jakob Julian Engel, and Tomas Hodan. HOT3D: Hand and object tracking in 3D from egocentric multi-view videos. In *CVPR*, 2025. 2, 7

Siddhant Bansal, Zhifan Zhu, Shashank Tripathi, Jiahe Zhao, Michael J. Black, and Dima Damen. Towards in-the-wild egocentric 3d hand-object pose estimation. In *ECCV*, 2026. 3, 23, 24

Chen Bao, Jiarui Xu, Xiaolong Wang, Abhinav Gupta, and Homanga Bharadhwaj. HandsOnVLM: Vision-language models for hand-object interaction prediction. *TMLR*, 2025. 3

Wentao Bao, Lele Chen, Libing Zeng, Zhong Li, Yi Xu, Junsong Yuan, and Yu Kong. Uncertainty-aware state space transformer for egocentric 3D hand trajectory forecasting. In *ICCV*, 2023. 3, 9

Homanga Bharadhwaj and Yash Jangir. MotionForesight: Re-purposing video models for future 3D scene-flow prediction. *arXiv preprint arXiv:2607.16192*, 2026. 3, 9

Homanga Bharadhwaj, Abhinav Gupta, Vikash Kumar, and Shubham Tulsiani. Towards generalizable zero-shot manipulation via translating human interaction plans. In *ICRA*, 2024a. 3

Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2Act: Predicting point tracks from internet videos enables generalizable robot manipulation. In *ECCV*, 2024b. 3

Homanga Bharadhwaj, Debidatta Dwibedi, Abhinav Gupta, Shubham Tulsiani, Carl Doersch, Ted Xiao, Dhruv Shah, Fei Xia, Dorsa Sadigh, and Sean Kirmani. Gen2Act: Human video generation in novel scenarios enables generalizable robot manipulation. In *CoRL*, 2025. 3

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. SAM 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025. 6, 21

Junuk Cha, Jihyeon Kim, Jae Shin Yoon, and Seungryul Baek. Text2HOI: Text-guided 3D motion generation for hand-object interaction. In *CVPR*, 2024. 3

Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, Jan Kautz, and Dieter Fox. DexYCB: A benchmark for capturing hand grasping of objects. In *CVPR*, 2021. 7

Delong Chen, Tejaswi Kasarla, Yejin Bang, Mustafa Shukor, Willy Chung, Jade Yu, Allen Bolourchi, Theo Moutakanni, and Pascale Fung. Action100M: A large-scale video action dataset. *arXiv preprint arXiv:2601.10592*, 2026a. 6, 21

Mingfei Chen, Yifan Wang, Zhengqin Li, Homanga Bharadhwaj, Yujin Chen, Chuan Qin, Ziyi Kou, Yuan Tian, Eric Whitmire, Rajinder Sodhi, Hrvoje Benko, Eli Shlizerman, and Yue Liu. EgoMAN: Interaction-structured reasoning for egocentric 3D hand trajectory prediction. In *ECCV*, 2026b. 3

Xingyu Chen, Fu-Jen Chu, Pierre Gleize, Kevin J Liang, Alexander Sax, Hao Tang, Weiyao Wang, Michelle Guo, Thibaut Hardin, Xiang Li, et al. SAM 3D: 3Dfy anything in images. *arXiv preprint arXiv:2511.16624*, 2025. 6, 21

Zerui Chen, Rolandos Alexandros Potamias, Shizhe Chen, Jiankang Deng, Cordelia Schmid, and Stefanos Zafeiriou. HO-Flow: Generalizable hand-object interaction generation with latent flow matching. In *ECCV*, 2026c. 3

Sammy Christen, Muhammed Kocabas, Emre Aksan, Jemin Hwangbo, Jie Song, and Otmar Hilliges. D-Grasp: Physically plausible dynamic grasp synthesis for hand-object interactions. In *CVPR*, 2022. 3

Christopher Clark, Jieyu Zhang, Zixian Ma, Jae Sung Park, Mohammadreza Salehi, Rohun Tripathi, Sangho Lee, Zhongzheng Ren, Chris Dongjoo Kim, Yinuo Yang, et al. Molmo2: Open weights and data for vision-language models with video understanding and grounding. *arXiv preprint arXiv:2601.10611*, 2026. 6, 21

Setareh Cohan, Guy Tevet, Daniele Reda, Xue Bin Peng, and Michiel van de Panne. Flexible motion in-betweening with diffusion models. In *ACM SIGGRAPH*, 2024. 3

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *CVPR*, 2017. 16

Valentin De Bortoli, Alexandre Galashov, J. Swaroop Guntupalli, Guangyao Zhou, Kevin Patrick Murphy, Arthur Gretton, and Arnaud Doucet. Distributional diffusion models with scoring rules. In *ICML*, 2025. 6

Zicong Fan, Omid Taheri, Dimitrios Tzionas, Muhammed Kocabas, Manuel Kaufmann, Michael J. Black, and Otmar Hilliges. ARCTIC: A dataset for dexterous bimanual hand-object manipulation. In *CVPR*, 2023. 24

Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981. 21

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 2007. 6

Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. HOnnotate: A method for 3D annotation of hand and object poses. In *CVPR*, 2020. 7

Yana Hasson, Gül Varol, Cordelia Schmid, and Ivan Laptev. Towards unconstrained joint hand-object reconstruction from RGB videos. In *3DV*, 2021. 3

Masashi Hatano, Zhifan Zhu, Hideo Saito, and Dima Damen. The invisible EgoHand: 3D hand forecasting through EgoBody pose estimation. *arXiv preprint arXiv:2504.08654*, 2025. 9

Peter J Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 1964. 7, 21

Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. CoTracker3: Simpler and better point tracking by pseudo-labelling real videos. In *ICCV*, 2025. 6, 21

Taein Kwon, Bugra Tekin, Jan Stühmer, Federica Bogo, and Marc Pollefeys. H2O: Two hands manipulating objects for first person interaction recognition. In *ICCV*, 2021. 7

Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. EPnP: An accurate O(n) solution to the PnP problem. *IJCV*, 2009. 21

Jiaman Li, Alexander Clegg, Roozbeh Mottaghi, Jiajun Wu, Xavier Puig, and C. Karen Liu. Controllable human-object interaction synthesis. In *ECCV*, 2024. 3

Muchen Li, Sammy Christen, Chengde Wan, Yujun Cai, Renjie Liao, Leonid Sigal, and Shugao Ma. LatentHOI: On the generalizable hand object motion generation with latent hand diffusion. In *CVPR*, 2025a. 3

Zhengqi Li, Richard Tucker, Forrester Cole, Qianqian Wang, Linyi Jin, Vickie Ye, Angjoo Kanazawa, Aleksander Holynski, and Noah Snavely. MegaSaM: Accurate, fast and robust structure and motion from casual dynamic videos. In *CVPR*, 2025b. 6

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *ICLR*, 2023. 5

Shaowei Liu, Subarna Tripathi, Somdeb Majumdar, and Xiaolong Wang. Joint hand motion and interaction hotspots prediction from egocentric videos. In *CVPR*, 2022. 3

Yun Liu, Haolin Yang, Xu Si, Ling Liu, Zipeng Li, Yuxiang Zhang, Yebin Liu, and Li Yi. TACO: Benchmarking generalizable bimanual tool-action-object understanding. In *CVPR*, 2024. 7

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 18

Junyi Ma, Wentao Bao, Jingyi Xu, Guanzhong Sun, Yu Zheng, Erhang Zhang, Xieyuanli Chen, and Hesheng Wang. Uni-Hand: Universal hand motion forecasting in egocentric views. *PAMI*, 2026. 9

Jeongwan On, Kyeonghwan Gwak, Gunyoung Kang, Junuk Cha, Soohyun Hwang, Hyein Hwang, and Seungryul Baek. BIGS: Bimanual category-agnostic interaction reconstruction from monocular videos via 3D gaussian splatting. In *CVPR*, 2025. 3

Lorenzo Pacchiardi, Rilwan Adewoyin, Peter Dueben, and Ritabrata Dutta. Probabilistic forecasting with generative networks via scoring rule minimization. *JMLR*, 2024. 6

Chaoyi Pan, Changhao Wang, Haozhi Qi, Zixi Liu, Homanga Bharadhwaj, Akash Sharma, Tingfan Wu, Guanya Shi, Jitendra Malik, and Francois Hogan. SPIDER: Scalable physics-informed dexterous retargeting. In *IROS*, 2026. 2, 9

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023. 16

Romain Pic, Clément Dombry, Philippe Naveau, and Maxime Taillardat. Proper scoring rules for multivariate probabilistic forecasts based on aggregation and transformation. *Advances in Statistical Climatology, Meteorology and Oceanography*, 2025. 6

Rolandos Alexandros Potamias, Jinglei Zhang, Jiankang Deng, and Stefanos Zafeiriou. WiLoR: End-to-end 3D hand localization and reconstruction in-the-wild. In *CVPR*, 2025. 6, 21

Aditya Prakash, Richard Li, David Forsyth, and Saurabh Gupta. Bimanual 3D hand motion and articulation forecasting in everyday images. *arXiv preprint arXiv:2510.06145*, 2025. 9

Patrick Rim, Kevin Harris, Braden Copple, Shangchen Han, Xu Xie, Ivan Shugurov, Sizhe An, He Wen, Alex Wong, Tomas Hodan, and Kun He. SHOW3D: Capturing scenes of 3D hands and objects in the wild. In *CVPR*, 2026. 7

Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics*, 2017. 4, 21

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025. 5, 16

Rustin Soraki, Homanga Bharadhwaj, Ali Farhadi, and Roozbeh Mottaghi. ObjectForesight: Predicting future 3D object trajectories from human videos. In *ECCV*, 2026. 3, 9

Neerja Thakkar, Tara Sadjadpour, Jathushan Rajasegeran, Shiry Ginosar, and Jitendra Malik. Poly-autoregressive prediction for modeling interactions. In *CVPR*, 2025. 3

Neerja Thakkar, Shiry Ginosar, Jacob C Walker, Jitendra Malik, João Carreira, and Carl Doersch. Forecasting motion in the wild. In *ECCV*, 2026. 3

Jikai Wang, Qifan Zhang, Yu-Wei Chao, Bowen Wen, Xiaohu Guo, and Yu Xiang. HO-Cap: A capture system and dataset for 3D reconstruction and pose tracking of hand-object interaction. *arXiv preprint arXiv:2406.06843*, 2024. 7

Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. MoGe: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In *CVPR*, 2025a. 6, 21

Shibo Wang, Haonan He, Maria Parelli, Christoph Gebhardt, Zicong Fan, and Jie Song. MagicHOI: Leveraging 3D priors for accurate hand-object reconstruction from short monocular video clips. In *ICCV*, 2025b. 3

Sirui Xu, Zhengyuan Li, Yu-Xiong Wang, and Liang-Yan Gui. InterDiff: Generating 3D human-object interactions with physics-informed diffusion. In *ICCV*, 2023. 3

Yufei Ye, Abhinav Gupta, Kris Kitani, and Shubham Tulsiani. G-HOP: Generative hand-object prior for interaction reconstruction and grasp synthesis. In *CVPR*, 2024. 3

Yufei Ye, Yao Feng, Omid Taheri, Haiwen Feng, Shubham Tulsiani, and Michael J. Black. Predicting 4D hand trajectory from monocular videos. In *3DV*, 2026a. 3

Yufei Ye, Jiaman Li, Ryan Rong, and C. Karen Liu. WHOLE: World-grounded hand-object lifted from egocentric videos. In *CVPR Findings*, 2026b. 3

Chengbo Yuan, Chuan Wen, Tong Zhang, and Yang Gao. General flow as foundation affordance for scalable robot learning. In *CoRL*, 2024. 9

Yuanwen Yue, Damien Robert, Jianyuan Wang, Sunghwan Hong, Jan Dirk Wegner, Christian Rupprecht, and Konrad Schindler. LitePT: Lighter yet stronger point transformer. In *CVPR*, 2026. 5, 16

Xinyu Zhan, Lixin Yang, Yifei Zhao, Kangrui Mao, Hanlin Xu, Zenan Lin, Kailin Li, and Cewu Lu. OAKINK2: A dataset of bimanual hands-object manipulation in complex task completion. In *CVPR*, 2024. 2, 3, 7

Jianing Zhang, Chenhao Zheng, Yajun Yang, Max Argus, Rustin Soraki, Winson Han, Taira Anderson, Chun-Liang Li, Shuo Liu, Jiafei Duan, Zhongzheng Ren, Jieyu Zhang, and Ranjay Krishna. MolmoMotion: Forecasting point trajectories in 3D with language instruction. *arXiv preprint arXiv:2606.18558*, 2026. 3, 9

Junsheng Zhou, Jinsheng Wang, Baorui Ma, Yu-Shen Liu, Tiejun Huang, and Xinlong Wang. Uni3D: Exploring unified 3D representation at scale. In *ICLR*, 2024. 5, 17

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *CVPR*, 2019. 4

Supplementary Material

Contents

A Model and Training Details 16

B Extraction Pipeline Details 21

B.1 Pipeline Implementation 21

B.2 Refinement Details 22

B.3 Extraction Quality 23

C Prediction Benchmark Details 24

D Model Ablations 25

E Additional Model Results 27

E.1 Task Specialists 27

F Limitations 27

A Model and Training Details

Sections 3.2, 3.3, and 4.1 provide additional details on the trajectory representation, model architecture, training objectives, and optimization. Tables 4 and 5 summarize the model and training settings and loss weights.

Representation. We sample windows every 2.5 seconds and retain the original timestamps. Each window contains up to 150 frames from 30 Hz recordings or about 50 frames from 10 Hz extracted videos. We pad shorter windows and exclude padding and absent entities from attention and losses.

When gravity is known, we rotate the coordinate system so that gravity points downward. If the object is unobserved in the last observed frame, we center on the mean observed hand position. If no hand is observed in that frame, we center on the object position.

Δ-codec. We compute rotation changes by subtracting consecutive 6-D rotation vectors, including those for finger joints. Relation tokens describe each frame directly, without temporal differences. Each standardized absolute-state input is paired with a flag that is 1 when the state is observed and 0 otherwise. A linear layer initialized to zero maps these inputs to a vector that we add to the state embedding. Both inputs are zero at unobserved frames.

A gap is a sequence of consecutive unobserved states during which the entity remains present. Corrections use adjacent observed poses from the same uninterrupted interval. If only one end of a gap is observed, we apply its translation and rotation corrections throughout the gap. With neither end observed, the predicted poses remain unchanged. Rotation corrections also include finger joints.

Flow-matching loss. We compute $\mathcal{L}_{\mathrm{FM}}$ separately for hand translation, wrist rotation, finger articulation, object translation, object rotation, and hand-object relations. Within each window, each term averages squared velocity errors over its channels and valid masked frames. Hand and relation terms give equal weight to the left and right sides when both have valid targets. We combine the terms using a weighted mean with the weights in Table 5, then average across windows. Terms and windows without valid targets are excluded. This prevents longer windows or larger channel groups from receiving extra weight. For losses on decoded trajectories, we estimate clean tokens as $\hat{\mathbf{x}}_1 = \mathbf{x}t + (1-t)v\theta$ and decode them into poses.

Architecture. We use a 12-block transformer with 512-dimensional embeddings and eight attention heads, following the attention pattern in Section 3.2. Separate two-layer MLPs embed hand states, object states, and relations, and linear output layers predict flow velocities. Both hands use the same input MLP and output layer. The two relation tokens share a separate input MLP and output layer. Each attention and feed-forward layer receives the flow time and the average time between frames through AdaLN-Zero (Peebles & Xie, 2023). A separate MLP combines the three state tokens to predict fingertip contact. We train it with binary cross-entropy, defining contact as a fingertip-to-object distance below 5 mm. Table 4 gives the remaining settings.

Each entity's confidence features describe motion smoothness, stability of fingertip-to-object distances, presence in nearby frames, and source quality. For relation tokens, we concatenate the hand and object confidence vectors. To avoid revealing masked poses, we reset smoothness to 0.5 within 0.1 seconds of the entity's masked frames. Fingertip-distance scores follow the same rule for all hand and object poses used to compute them.

Conditioning. We build the scene point cloud from the last observed frame. We exclude hand and object pixels using masks from dataset annotations or projected hand and object meshes. We crop the remaining background within 0.5 m of the hand-object midpoint, expanding this radius to 1 or 2 m if fewer than 500 points remain. Farthest-point sampling selects 4,096 points, which we express in the same centered coordinates as the trajectories.

Each point has 12 geometric features describing its position, surface normal, distances to the hands and object, height above the support plane, normal orientation relative to vertical, curvature, and log density. We add DINOv3 image features (Siméoni et al., 2025), reduced to 32 dimensions using a PCA projection shared across clips. LitePT-S (Yue et al., 2026), pretrained on ScanNet (Dai et al., 2017), encodes these features. Eight learned queries pool the encoded points into eight scene tokens through two attention blocks. We train the input layer, output projection, and pooling blocks while keeping the rest of the encoder fixed. During training, we zero the image features in 20% of windows.

For object shape, we precompute frozen Uni3D-g features (Zhou et al., 2024) from 10,000 points sampled uniformly over the mesh surface, with positions normalized to a unit sphere and surface normals included. A trainable projection maps these features to four object-shape tokens. Separate tokens encode the object's log scale and each hand's MANO shape. We mask the shape token of any hand absent throughout the window.

Training masks. We sample observation patterns with the probabilities in Table 3. For forecasting, we randomly choose the number of initial frames to observe. For goal-conditioned generation, we observe the first 1 to 5 frames and the last 1 to 3 frames, choosing both counts randomly. For temporal infilling, we sample the gap fraction and starting position uniformly while keeping the first and last frames observed.

Random-block masking selects each entity independently with probability 0.5. For each selected entity, we randomly choose how many consecutive frames to mask, from one frame to all but one valid frame. We then randomly choose where this segment starts. We keep the first frame observed and ensure that at least one hand or object pose is masked later in the window. We exclude padding and absent entities and convert the resulting observation masks to Δ-codec masks before training.

Trajectory losses. We supervise decoded wrist and object positions with squared error. Rotation losses cover the wrists, object, and 15 finger joints of each hand. Each rotation error is $\theta^2/2$, where θ is the angle between the predicted and reference rotations. These losses cover masked frames and observed frames whose decoded poses depend on predicted differences. We average each loss within a window, then across windows with valid targets. The smoothness loss uses wrist and object acceleration and jerk computed from the original timestamps. We penalize only increases in their squared magnitudes over the reference. Both penalties are computed within masked segments, without crossing into observed frames, and the jerk term has weight $1/30$. We regularize finger articulation using joint limits and the MANO pose prior. Joint-limit penalties apply only when the predicted violations exceed those in the training target. We use the MANO pose prior similarly, penalizing only increases over the target's prior penalty. Table 4 lists their settings.

For fingertip-distance supervision, we minimize absolute errors in fingertip-to-object distances, considering only fingertips within 10 cm of the object in the reference. For each hand and training window, we identify frames where the hand or object pose depends on a model prediction. We use up to 16 of these frames, always including the last. We apply the attachment loss in Equation 3 when reference contact lasts for at least two consecutive frames. For each hand, we use up to 48 frames from the longest contact periods in the window. The velocity term also supervises how the predicted motion connects to observed motion while contact continues.

For sampled trajectories, each energy-score term in Equation 4 is

$$\mathrm{ES}(\{z_i\}_{i=1}^{M}, z_Y) = \frac{1}{M}\sum_i \|z_i - z_Y\|_2 - \frac{1}{2M(M-1)}\sum_{i\neq j}\|z_i - z_j\|_2, \tag{5}$$

where $z_i = \phi(X_i)$ and $z_Y = \phi(Y)$ describe the sampled and target motion. The first score uses position changes at three selected frames and the final rotation change. The second uses position changes at every predicted frame. We measure these changes from the target pose at the last observed frame. We also include changes in wrist position relative to the object whenever the hand or object is being predicted. We scale the position changes in the second score to account for differences in the number of predicted frames. Table 4 gives the rotation encoding and scaling factors. We also penalize predicted objects that extend more than 10 mm below nearby supporting surfaces, such as tabletops.

Training. We sample windows using the source and label-quality weights in Table 4. To balance supervision between the left and right hands, we increase the weights of windows containing a left hand by at most a factor of 10. The sampling probability for extracted videos stays at approximately 23.5% for the first 20,000 steps, decreases linearly to 2% by step 32,000, and remains there until training ends.

With probability 0.5, we rotate the hands, object, and scene together around the gravity axis before centering. Independently, with probability 0.5, we sample 30 Hz windows at 30, 15, or 10 Hz, choosing each rate equally. We retain the original timestamps and five-second window boundaries, so the motion timing stays unchanged.

We train for 40,000 optimizer steps with AdamW (Loshchilov & Hutter, 2019) and evaluate the final checkpoint. Table 4 lists the optimizer settings, learning-rate schedule, batch size, and compute details.

Inference. We initialize masked state and relation tokens with standard Gaussian noise and update them together using 10 equally spaced Euler steps from $t = 0$ to $t = 1$. The Δ-codec converts the generated state tokens into poses, after which we add back the translation removed during centering. To generate multiple trajectories, we repeat this procedure with independent initial noise while keeping the observations and conditioning fixed.

Table 3: **Observation patterns for training and evaluation.** T_v denotes the number of valid frames. †Within goal-conditioned training, we provide goals for all entities, only the hands, or only the object with probabilities 0.5, 0.25, and 0.25, respectively. Entities without a goal are observed only in the first frame.

Mode	Observed in training	Observed at evaluation	Training probability
Future prediction	first K frames, $K \sim \mathrm{Unif}\{1, \ldots, T_v - 1\}$	first 0.5 s	0.19
Hands → object	hands throughout; object at first frame	same	0.19
Object → hands	object throughout; hands at first frame	same	0.19
Goal-conditioned†	first 1 to 5 and last 1 to 3 frames	first and last frame	0.19
Temporal infilling	all but one gap of 20 to 50%	all but the central third	0.19
Random blocks	all but one block per selected entity	training only	0.05

Loss weights. We set the flow-matching component weights and selected auxiliary loss weights using gradient norms measured before training on batches from the annotated datasets. We rescale these weights toward chosen gradient contributions relative to the flow-matching loss. Table 5 lists the resulting weights. All remain fixed during training except the contact-classifier weight, which increases linearly from approximately zero to 0.1 over the first 5,000 steps.

Table 4: **Model and training settings.** Dataset sizes are shown as training trajectories / five-second windows.

Architecture	
Token dimensions	Hand state 99; object state 9; relation 14
Transformer	12 blocks; width 512; 8 heads; feed-forward width 2048; GELU; dropout 0.1
Scene output projection	Trainable, $72 \to 64$
Contact classifier	MLP, $1536 \to 512 \to 10$
Parameters	103.3M trainable; 12.7M frozen
Training data	
Dataset sizes	Action100M 62,562/109,346; TACO 4,089/6,267; DexYCB 2,280/2,280; OakInk2 2,213/9,938; SHOW3D 1,638/3,552; HOT3D 1,016/1,016; H2O 191/930; HO3D 173/867; HO-Cap 150/465; total 74,312/134,661
Data splits	Grouped by recording
Source weighting	Initial total sampling weight per dataset $\propto N_s^{1/4}$, where N_s is its number of training windows
Label-quality weights	Motion capture 3; fitted 2; landmark-fitted 1.5; extracted 1
Rotation augmentation	Zero-mean Gaussian angle with standard deviation 0.35 radians
Optimization	
Optimizer	AdamW, $\beta_1 = 0.9$, $\beta_2 = 0.999$
Learning rate	Linear warmup to 10^{-4} over 1,000 steps; cosine decay to 10^{-5} at step 40,000
Weight decay	0.01, excluding biases, normalization parameters, embeddings, learned queries, and other one-dimensional parameters
Gradient clipping	Global norm 1.0
Batch size	512 windows globally; 32 per GPU
Compute	16 H100 GPUs across two nodes; approximately two days
Precision	bfloat16 mixed precision; float32 for the point-cloud backbone, augmentation, and geometric losses
Random seed	42
Loss settings	
Finger-joint limits	Twist around the rest bone axis: 25° at finger bases and 15° elsewhere; swing: 135°. Compare each joint's penalty with its training target.
MANO prior normalization	Divide each increase in a squared standardized PCA coefficient by the larger of the target's squared coefficient and 1.
Energy-score rotation encoding	Flatten $0.3(R_{\mathrm{ref}}^\top R_{\mathrm{end}} - I)/\sqrt{2}$, where R_{ref} is the target rotation at the last observed frame and R_{end} is the rotation at the last predicted frame.
Energy-score position scaling	For the score using every predicted frame, scale each position-change sequence by $\sqrt{3/n}$, where n is its number of valid predicted frames.
Support-surface selection	Surface tilt at most 30° from horizontal. Each object vertex uses the nearest eligible scene point if their horizontal distance is at most 40 mm.
Penetration averaging	Mean of the eight largest vertex penalties per frame, then mean over predicted frames and samples within each window.

Table 5: **Training loss weights.** The first six rows give weights within the flow-matching loss; the remaining rows give coefficients for the auxiliary losses.

Component	Weight
Flow matching: hand translation	0.884
Flow matching: wrist rotation	0.738
Flow matching: finger articulation	1.112
Flow matching: object translation	0.730
Flow matching: object rotation	0.306
Flow matching: hand-object relations	0.185
Decoded translation	1.0
Decoded wrist and object rotation	0.368
Decoded finger-joint rotation	0.422
Smoothness excess	3.2×10^{-4}
Fingertip distance	0.335
Attachment position	0.178
Attachment rotation	0.041
Attachment velocity	0.121
Joint-limit excess	3.35×10^{-3}
MANO prior excess	5.0×10^{-4}
Energy score	1.0
Support penetration	1.0
Contact classifier (final)	0.1

B Extraction Pipeline Details

B.1 Pipeline Implementation

We describe the exported trajectories, processing steps, and quality checks for the video extraction pipeline introduced in Section 3.4.

Exported trajectories. Each exported trajectory contains frames sampled at approximately 10 Hz, with their original video timestamps. It includes MANO parameters (Romero et al., 2017) for up to two hands, per-frame validity flags for each hand, and the rotation and translation of a reconstructed object mesh. The export also includes an estimated gravity direction when available.

Clip selection and validation. Candidate clips come from Action100M's timestamped action captions (Chen et al., 2026a). Molmo 2 (Clark et al., 2026) identifies the manipulated object from each caption without a predefined object list. We discard candidates for which it identifies no object. The same model then checks whether the video shows a hand manipulating any object. For clips that pass, Molmo 2 checks one frame per second to find when the interaction occurs. We use these times to choose frames for locating the object. We split clips at detected shot boundaries and process shots lasting at least 4 seconds independently.

Hand and object estimation. We use WiLoR (Potamias et al., 2025) to obtain MANO hand estimates. We remove low-confidence detections and short-lived tracks, rejecting clips with no remaining hand track. To locate the object, we ask Molmo 2 (Clark et al., 2026) to point to it. Prompts use the object name from the caption. For generic names such as "object," we instead ask for "the object held in the hand." We discard points that lie far from the others. When hand locations are available, we also check that the remaining group is near a hand. If these checks fail, we try another frame. The accepted points initialize SAM 3 (Carion et al., 2025), which tracks the object mask forward and backward through the clip.

SAM 3D (Chen et al., 2025) then reconstructs the object mesh from one frame. To select this frame, we favor larger object masks. We give lower priority to masks that are much larger than those in nearby frames, split into several parts, or unusually irregular in shape. At other sampled frames, we reuse the reconstructed mesh and run SAM 3D again to estimate the object's pose and scale. If this step fails, we run full reconstruction for that frame.

Object motion estimation. We track points in the object mask with CoTracker3 (Karaev et al., 2025) and lift them into 3D using depth from MoGe (Wang et al., 2025a). We initialize object poses with RANSAC PnP (Fischler & Bolles, 1981; Lepetit et al., 2009), refine them, and remove tracks with high reprojection error. We then refine the poses and the points' positions within the object together, keeping camera poses fixed. A Huber loss (Huber, 1964) measures agreement with the image tracks, while additional penalties encourage smooth rotations and preserve the estimated 3D shape. We combine the fitted rotations with SAM 3D's estimates of the object center. If no fit is available or the median per-frame reprojection error exceeds 12 pixels, we use SAM 3D's rotation estimates instead.

Joint optimization. We refine the hand and object poses together while keeping hand shape and size fixed. To reduce jitter, we constrain adjustments to object motion and finger poses to change smoothly over time. We also refine the object size estimated from MoGe depth, using a single scale factor for the whole clip.

The objective matches the projected object to its mask, preserves fingertip, finger, and palm contact, and penalizes interpenetration. Additional terms discourage large corrections to the initial estimates, abrupt motion, and unnatural finger poses. We also penalize changes in relative hand-object position and changes in relative rotation during grasps. We penalize changes to reliable hand estimates more strongly.

We use the Adam optimizer with a cosine learning-rate schedule. We first align the wrists in the opening frames and keep those poses fixed. We then refine object and wrist motion before adjusting the finger poses. A joint stage refines these object, wrist, and finger estimates together. Finally, we adjust wrist and finger poses in frames with low tracking confidence and poor hand-object alignment. The object stays fixed, while nearby reliable frames provide target finger positions relative to the

palm. After joint optimization, we correct object rotation during occlusions, apply a learned depth-bias correction, and smooth hand jitter (Appendix B.2).

Object scale and depth correction. Reconstruction errors can make an object change size and depth across frames while still matching its image mask. We apply a correction when the 95th-percentile object size is at least 1.35 times the 5th-percentile size. We first rescale each frame's mesh to the clip's median object size. We then compare its projected area with the observed mask area to estimate a smooth depth correction. For this estimate, we exclude frames where hands cover at least half the projected object. We reject corrections with too few reliable observations or increased depth variation. When correcting the object, we also shift any hand in contact with it to maintain that contact.

Interaction windows. We identify interaction frames where the object is visible and a tracked hand lies within 5 cm of its surface. We group these frames into intervals, merging gaps of at most 2 seconds. Frames with abrupt changes in estimated object motion are excluded, and windows cannot cross them. We add up to 0.75 seconds before and after each interval to include approach and release motion. Each window must last at least 2.5 seconds and contain at least eight sampled interaction frames. We prevent windows from overlapping and export each as a separate trajectory.

Final pose refinement. Within each window, we adjust the object's pose to improve overlap between its projected silhouette and the observed mask. We limit rotation and translation corrections to 35° and 0.30 m and smooth them over five sampled frames. We identify grasps when a hand stays within 4 cm of the object for at least four consecutive sampled frames. During grasps, each correction moves the hand and object together, preserving their relative geometry. Outside grasps, we accept a correction only if it improves mask IoU by at least 0.03.

We then smooth the hand's rotation relative to the object over seven sampled frames during grasps, without changing finger articulation. Finally, we adjust hand positions by at most 8 mm to reduce any remaining hand-object penetration.

Gravity estimation. We estimate gravity by finding a supporting surface, like a table or floor, in MoGe depth maps from up to eight evenly spaced frames per window. We check that these surfaces face the same direction across frames. Candidate surfaces must lie below most hand, object, and scene points. We rank them by scene coverage and proximity to the interaction, favoring surfaces that the object approaches closely. To avoid confusing walls with supporting surfaces, we assume the camera is tilted sideways by no more than 45° and points less than 30° above horizontal. We exclude surfaces that appear almost as a line in the image or are viewed from directly above. Gravity points perpendicular to the selected surface, away from the interaction. If no surface passes these checks, we leave gravity unspecified.

Quality filtering. We check that each reconstructed trajectory matches the video by comparing the projected object's outline and size with its tracked mask. We also check for changes in object shape or size, abrupt motion, and missing hand-object contact. A tracked hand must remain within 5 cm of the object in at least 25% of sampled frames. Additional checks detect excessive hand-object penetration, shot boundaries, and object masks that track the hand itself. We also check that estimated contact points project inside the object mask before the final pose correction, so that correction cannot create its own evidence of contact. Motion checks use the corrected trajectory.

For windows that pass, we ask Molmo 2 to "point to the object held in the hand" in up to five frames, collecting six responses per frame without providing an object name. We reject windows when more than 65% of responses contain no point or the predicted points are too far from the tracked object mask. These decisions require responses from at least two frames. Because this check can still accept a held object that is not the intended manipulation target, we assign retained trajectories an interaction-fidelity score in $[0.7, 1]$ based on contact and motion agreement. We include this score in the model's confidence input to account for the remaining uncertainty.g

B.2 Refinement Details

Object rotation. When the object is occluded for at least three consecutive sampled frames, we use the grasping hand's motion to guide its rotation. We apply this correction only when contact is stable and the camera rotates slowly. During the occlusion, we make the object rotate more closely with the hand. We gradually increase the correction as the occlusion begins and reduce it as the occlusion

ends. The corrected orientation carries forward into later frames rather than returning to the original estimate. We reject corrections that introduce large rotation jumps.

Bias denoiser. We train a separate denoiser to recover clean trajectories from simulated extraction errors. Training uses 11,568 windows of 50 frames from DexYCB and HOT3D, sampled at 10 Hz. We remove the 129 segments that also appear in the extraction benchmark from the training and validation sets. We deliberately alter the clean trajectories to reproduce errors observed on this benchmark. For example, we place hands and objects too close to or too far from the camera, disturb their positions over several frames, and gradually change the object's orientation. We also add larger errors over short intervals to simulate unreliable estimates during occlusion. The denoiser learns to recover the original trajectories from these altered versions.

The denoiser receives the hand and object poses at each frame, including finger articulation. It predicts corrections that we add to the standardized input states. The model uses an eight-layer bidirectional transformer with width 384, six attention heads, and feedforward width 1024. We train for 15,000 steps with batch size 256, learning rate 2×10^{-4}, and 500 warmup steps. The loss measures squared reconstruction errors, with weights of 10 for translation, 3 for rotation, and 1 for finger articulation. A velocity loss, weighted by 0.5, encourages the recovered motion to match the clean trajectory.

Applying the correction. We use the denoiser's predictions to estimate how far each hand and the object should move toward or away from the camera. We average these corrections over the trajectory and apply the same shift at every frame. For the object and any contacting hands, we combine their shifts and move them together to preserve their relative positions. Finally, we constrain finger rotations to anatomical limits and smooth wrist poses and finger rotations over three sampled frames.

Evaluation. Table 6 compares refinement variants on 30 DexYCB and 28 HOT3D sequences. Raw extraction uses trajectories before refinement. Unrestricted denoising applies all predicted pose corrections, while moving-average smoothing averages poses across neighboring frames. Full refinement combines constant position corrections from the denoiser with object-rotation correction during occlusion and hand smoothing. It reduces median wrist error from 106.4 to 88.6 mm on DexYCB and from 191.2 to 158.7 mm on HOT3D. Unrestricted denoising reduces depth bias but increases hand surface error and HOT3D object rotation drift. Moving-average smoothing gives inconsistent improvements across datasets.

Table 6: **Refinement results**. Hand metrics use the right hand. Wrist measures wrist position error; depth measures signed wrist depth error; surface measures hand vertex error after aligning the wrist positions. Object depth measures signed object-centroid depth error, and drift measures object rotation drift. Distances are in millimeters and rotations in degrees. Lower errors are better, except signed depth errors, which are best at zero.

	DexYCB				HOT3D				
Variant	wrist	depth	surface	obj. depth	wrist	depth	surface	obj. depth	drift
Raw extraction	106.4	−93.5	26.1	−56.2	191.2	153.3	67.4	69.7	30.1
Bias denoiser, unrestricted	97.6	−4.3	41.9	−12.3	162.5	72.5	93.6	19.6	43.5
Moving-average smoothing	99.2	−62.4	29.1	−48.6	205.7	173.2	72.1	75.7	32.2
Full refinement	88.6	−4.7	27.3	−23.8	158.7	78.6	70.2	19.4	30.1

B.3 Extraction Quality

We evaluate the extraction pipeline of Section 3.4 on 222 of the 237 EPIC-Contact test clips (Bansal et al., 2026) using the authors' released evaluator and fitted reference poses, without training on EPIC-Contact (Table 7). For frame-rate-dependent metrics, we compare against HOPformer predictions sampled at our approximately 10 Hz cadence. Our full pipeline achieves lower hand joint error (16.0 vs. 19.7 mm) and motion deviation (10.6 vs. 13.8 mm), but higher contact deviation (34.6 vs. 20.8 mm) and hand-object relative position error (71.8 vs. 64.6 mm). These results cover completed extractions; excluding failed clips can favor our reported scores. Absolute depth and hand-object placement remain limitations of the extracted trajectories.

Table 7: **EPIC-Contact reconstruction.** Acceleration errors are listed for hand/object. SR uses thresholds of 5%/10% of the object diameter. Published baselines use oracle object meshes; we reconstruct geometry from video given the object's name. Bold marks the better result in the matched-cadence comparison.

Method	Contact deviation ↓ (mm)	Relative position ↓ (mm)	Motion deviation ↓ (mm)	Acceleration ↓ (m/s^2)	Hand joint error ↓ (mm)	SR@0.05 ↑ (%)	SR@0.1 ↑ (%)
Published results							
ArcticNet-SF (Fan et al., 2023)	94.2	166.7	70.1	3.9 / 5.3	44.8	16.5	55.9
JointTransformer (Abou Zeid, 2023)	30.1	78.6	20.0	3.1 / 9.5	22.9	17.6	56.9
HOPformer (Bansal et al., 2026)	20.7	65.8	11.4	2.5 / 4.1	19.9	29.8	69.7
Evaluation at approximately 10 Hz							
HOPformer	**20.8**	**64.6**	13.8	3.9 / 7.9	19.7	**29.6**	69.1
Ours	34.6	71.8	**10.6**	**3.7 / 3.8**	**16.0**	28.5	**74.9**

C Prediction Benchmark Details

Benchmark construction. We select all benchmark data from sequences excluded from training. The annotated subset contains 256 windows from 168 original sequences. We select 36 windows from DexYCB, 35 each from HO3D, HOT3D, OakInk2, SHOW3D, and TACO, 24 from HO-Cap, and 21 from H2O. Within each dataset, we sample from different sequences before selecting additional windows from the same sequence.

For Action100M videos, we refine the reconstructed wrist and object poses and apply automated quality checks. We then sample one window per video and use AI-assisted review to compare each reconstruction with the video at six sampled frames. This procedure selects 75 windows without hand-picking examples or using model predictions or prediction errors. These checks improve reference quality. We combine the annotated and extracted subsets into a single benchmark of 331 windows used across all tasks.

Trajectory scoring. We compute ADE only over generated poses where the hand or object is present. For FDE, the final pose must be predicted rather than provided as input. When the final pose is provided, we instead measure position error at the middle of the missing interval. Single-sample results use the first generated trajectory. For best-of-20, we choose the trajectory with the lowest total ADE across the predicted hands and object. We use that same trajectory for all associated metrics.

Aggregation. For each task, we first average scores within each dataset. We then average the annotated datasets, giving each dataset equal weight. We combine this result with the Action100M average, so annotated data and internet videos each contribute half of the final score. We calculate these averages separately for each hand and the object. When reporting one hand score, we average the left and right hand scores. When summarizing multiple tasks, we give each task equal weight.

Additional metrics. We measure rotation error as the angle between predicted and reference orientations, in degrees. Finger articulation error averages this angle over the 15 finger joints of each hand. We count a fingertip as touching the object when it is within 5 mm of the surface, and compute contact F1 for windows containing reference contact. For closed object meshes, we measure how deeply hand vertices penetrate the object and the percentage of vertices inside it. We also count joint-limit violations introduced by the prediction, using the limits in Table 4. A violation counts only when the predicted joint exceeds a limit but the reference joint does not. We report this count as a percentage of all joint poses evaluated across frames. These metrics use the aggregation procedure described above.

Interaction metrics. For Table 9, we evaluate frames where the hand and object are both present and at least one is predicted. We measure fingertip position error relative to the object on frames where the reference hand is in contact. Contact F1 uses a 10 mm distance threshold and allows predicted contact to occur up to two frames earlier or later than reference contact.

We identify a reference grasp when at least two fingertips stay within 10 mm of the object for at least three consecutive frames. During each grasp, we compare predicted and reference fingertip

movements relative to the object, measured from the start of the grasp. We average this error over all five fingertips. A frame is *held* when at least two predicted fingertips are within 10 mm of the object and the movement error is at most 10 mm. It is *slipping* when at least one fingertip is within 10 mm but the held condition fails, and *detached* when all fingertips are at least 20 mm away. Frames between these thresholds remain unclassified. We report the percentage of reference-grasp frames in each category. Object velocity error compares position changes between consecutive frames using their elapsed time.

Interaction-score aggregation. For each window, we average scores across all 20 generated trajectories and average the left and right hands when both are available. We then average windows within each task, separately for annotated data and Action100M. All annotated windows contribute equally, regardless of their source dataset. We average the tasks equally and combine the annotated and Action100M scores with equal weight. We estimate 95% confidence intervals for differences between models by resampling video segments 5,000 times, using the same resampled segments for both models.

Released baselines. For Table 2, we use the USST ResNet-18, ObjectForesight EPIC-DiT, and Uni-Hand HaMeR/DINOv2 checkpoints. The ForeHand4D checkpoint was trained on DexYCB, HOT3D, and H2O, while Uni-Hand's checkpoints were trained on robot demonstrations. We evaluate each method on windows with its required inputs. For methods that generate multiple samples, we use the same ADE selection rule as our model, considering only the hands or object that each method predicts. For methods that predict 3D point trajectories, we estimate object poses by fitting rotation and translation without changing object size. We interpolate predictions to match the reference timestamps.

D Model Ablations

We compare our main model with configurations that omit selected losses, inputs, or augmentation (Tables 8 and 9). All use the main model's training budget and optimizer settings. Evaluation follows Appendix C.

Losses and relation tokens. We evaluate one configuration without the attachment, decoded-contact, joint-limit, MANO-prior, and support-penetration losses, retaining clean-state and smoothness supervision. Another omits relation-token inputs and their flow-matching loss while retaining decoded relative-geometry supervision. The main model has lower fingertip error during contact than both configurations: 65.7 mm, compared with 72.2 mm without the physical-loss group and 70.4 mm without relation tokens. Its contact F1 is also higher at 68.2%, compared with 65.3% and 66.3%, respectively (Table 9).

Scene conditioning. Scene conditioning provides information about surrounding surfaces and obstacles. The configuration without the scene encoder omits both scene geometry and RGB-derived DINOv3 features, retaining object-shape inputs. Its object FDE is 116.1 mm for one sample and 80.1 mm at best-of-20, compared with 110.3 and 76.9 mm for the main model. Hand ADE is similar, while finger articulation and penetration errors are slightly lower (Table 8). Contact fingertip error is 67.6 mm, compared with 65.7 mm for the main model (Table 9).

Augmentation. The configuration without global-pose and temporal-stride augmentation has higher ADE and FDE across all reported trajectory metrics. Best-of-20 hand FDE is 102.7 mm, compared with 93.3 mm for the main model, and joint-limit violations occur in 10.8% of evaluated joint poses, compared with 9.9% (Table 8).

Table 8: **Model configurations.** We compare all configurations with the main model. (a) Trajectory errors in millimetres; $N{=}20$ denotes best-of-20 selection. (b) Physical metrics from one sample. All checkpoints are evaluated at 40,000 training updates.

(a) Trajectory accuracy

	Hands ADE↓		Object ADE↓		Hands FDE↓		Object FDE↓	
Model	$N{=}1$	$N{=}20$	$N{=}1$	$N{=}20$	$N{=}1$	$N{=}20$	$N{=}1$	$N{=}20$
Main model	86.7	60.2	64.9	42.0	128.5	93.3	110.3	76.9
Losses and relation tokens								
No physical-loss group	90.2	62.2	64.9	43.0	135.4	101.3	111.9	80.5
No relation tokens	91.2	62.4	67.9	43.7	132.7	98.8	113.4	78.8
Scene conditioning and augmentation								
No scene encoder	86.7	60.7	66.5	42.7	129.6	95.4	116.1	80.1
No pose or stride augmentation	90.2	62.2	67.4	43.3	134.5	102.7	115.9	81.2

(b) Physical quality

Model	Finger error (°)↓	Contact F1 (5 mm) (%)↑	Penetration depth (mm)↓	Penetrating vertices (%)↓	Joint-limit violations (%)↓
Main model	12.9	21.8	3.01	4.20	9.9
Losses and relation tokens					
No physical-loss group	12.9	20.5	2.98	3.94	10.2
No relation tokens	13.1	21.2	3.03	3.94	10.1
Scene conditioning and augmentation					
No scene encoder	12.8	21.7	2.93	4.14	9.9
No pose or stride augmentation	13.3	20.8	3.03	3.94	10.8

Table 9: **Interaction quality.** We average scores across 20 generated samples per window. Tip error uses reference-contact frames; held, slipping, and detached rates use reference-grasp frames. Contact F1 uses a 10 mm threshold and a two-frame tolerance. Model configurations match Table 8.

Model	Tip error (mm)↓	Contact F1 (10 mm) (%)↑	Held (%)↑	Slipping (%)↓	Detached (%)↓	Object vel. error (mm/s)↓
Main model	65.7	68.2	15.5	47.5	24.9	139.6
Losses and relation tokens						
No physical-loss group	72.2	65.3	14.3	44.5	29.1	139.3
No relation tokens	70.4	66.3	14.2	45.5	28.1	142.1
Scene conditioning and augmentation						
No scene encoder	67.6	67.3	15.6	46.6	26.0	140.0
No pose or stride augmentation	70.0	66.5	14.4	45.8	27.8	142.2

E Additional Model Results

We extend the evaluation in Section 4 with task-specialist comparisons. We follow the evaluation protocol in Appendix C.

E.1 Task Specialists

We train five specialists using the shared model's architecture, data, source-sampling schedule, objective, and training budget. Four specialize in forecasting, hands-to-object prediction, object-to-hands prediction, and temporal infilling; the fifth covers all three goal patterns. Each uses only its assigned task family during training, with corresponding normalization statistics. Tables 10 and 11 compare each specialist with the shared model on its trained tasks.

The shared model reduces single-sample forecasting FDE by 13 mm for objects and 12 mm for hands, and best-of-20 FDE by 2–5 mm (Table 10). For cross-entity prediction and temporal infilling, specialists have 2–6 mm lower single-sample errors, while best-of-20 errors differ by less than 3 mm. The shared model achieves lower best-of-20 FDE on both cross-entity tasks. The goal specialist has lower endpoint and mid-gap errors across all three goal patterns. Its largest gains occur when an entity's final pose is unspecified, reducing FDE by 20–35 mm for a single sample and 8–22 mm at best-of-20.

Table 10: **Comparison with task specialists.** Each specialist entry uses the model trained for the corresponding task; one goal specialist covers all three goal patterns. Evaluation and notation follow Table 1.

	Future prediction		Hands→obj.	Obj.→hands	Temporal infilling		Goal, all given		Goal, hands given		Goal, object given	
Entity	object	hands	object	hands	object	hands	object	hands	object	hands	object	hands
Error at	end	end	end	end	middle	middle	middle	middle	end	middle	middle	end
Stay-put	142.6	145.6	149.0	156.6	51.7	69.3	98.8	127.7	149.0	127.7	98.8	156.6
Constant velocity	627.1	819.3	–	–	–	–	–	–	–	–	–	–
Interpolation	–	–	–	–	38.2	49.0	76.0	92.3	–	92.3	76.0	–
Rigid carry	–	–	108.6	156.4	–	–	–	–	–	–	–	–
Specialist, $N{=}1$	159.1	169.3	54.5	92.6	37.6	48.6	84.8	97.9	91.5	100.0	88.1	109.1
Specialist, $N{=}20$	105.0	118.1	38.4	77.4	**22.8**	**31.3**	**56.3**	**69.2**	**69.4**	**71.3**	**60.1**	**84.2**
Motion capture only, $N{=}1$	159.0	185.6	71.6	132.9	39.0	50.0	91.2	101.9	146.1	110.3	96.4	166.9
Motion capture only, $N{=}20$	121.8	153.9	50.0	103.7	26.6	35.0	62.6	74.1	107.4	78.2	68.0	137.1
Ours, $N{=}1$	145.8	157.4	59.1	98.7	39.5	52.1	89.8	101.6	126.1	110.2	97.4	129.3
Ours, $N{=}20$	**102.8**	**113.2**	**36.6**	**75.1**	23.4	32.8	56.5	70.2	91.1	73.8	61.6	91.7

Table 11: **Task specialists under ADE.** Mean position error over generated frames (mm); hands average the left and right wrists. Models, task assignments, and evaluation follow Table 10. Bold and underline indicate the best and second-best results.

	Future prediction		Hands→obj.	Obj.→hands	Temporal infilling		Goal, all given		Goal, hands given		Goal, object given	
Entity	object	hands	object	hands	object	hands	object	hands	object	hands	object	hands
Stay-put	80.4	103.4	87.0	115.6	47.9	63.8	86.4	114.1	87.0	114.1	86.4	115.6
Constant velocity	287.7	398.5	–	–	–	–	–	–	–	–	–	–
Interpolation	–	–	–	–	27.9	37.5	55.1	73.8	–	73.8	55.1	–
Rigid carry	–	–	85.8	120.0	–	–	–	–	–	–	–	–
Specialist, $N{=}1$	99.3	119.0	34.9	75.7	26.1	35.9	64.6	84.9	87.1	88.4	66.6	103.6
Specialist, $N{=}20$	61.3	81.6	**24.0**	57.9	**17.0**	**25.2**	**42.9**	60.1	**57.5**	**62.8**	**45.2**	**72.2**
Motion capture only, $N{=}1$	101.7	125.3	45.1	98.4	27.3	40.5	66.1	88.1	104.9	95.6	70.4	126.3
Motion capture only, $N{=}20$	70.5	94.9	31.0	72.6	19.6	29.0	47.2	65.4	70.5	71.1	51.0	92.4
Ours, $N{=}1$	93.1	112.6	38.6	79.5	27.6	38.1	65.1	87.0	95.2	92.7	70.0	110.3
Ours, $N{=}20$	**58.9**	**78.7**	24.1	**56.2**	17.9	26.7	43.5	**59.7**	60.3	65.2	47.3	74.7

F Limitations

Our model is limited by the diversity and accuracy of its training data. Laboratory datasets provide 3D annotations but cover a limited range of objects, environments, and interactions, which may restrict generalization to everyday activities. We broaden this coverage with internet videos, but recovering reliable 3D hand-object motion from them remains difficult. Our pipeline relies on pretrained depth and pose estimation models. Errors in these estimates, particularly during occlusion, can distort hand-object placement and contact. We reduce these errors through joint refinement

and quality filtering, but uncertain depth and hidden finger poses remain difficult to resolve. Filtering may also favor interactions that are easier to reconstruct, limiting the diversity of the retained training data.